AMENDMENTS TO THE SPECIFICATION

Please replace the paragraph on page 14, line 27- page 16, line 10 with the following amended paragraph:

Turning to Fig. 8, an exemplary distinct query count methodology 800 is illustrated in accordance with an aspect of the subject invention. Distinct count methodology 800 describes a method of determining the number of distinct or unique records in a single partition. Such methodology can be adapted to execution of a distinct count query over a plurality of partitions in parallel as described hereinafter. At 810, a count is initialized to zero. The count variable can be utilized to keep track of the number of distinct measures in the partition. At 812, the partition is examined to determine if it contains values or it is empty. If the partition is empty the process proceeds to 832, wherein the count is returned. Since no records or measures are available to be examined if the partition is empty the value returned is zero. If, however, the partition is not empty, the first value of the partition is read at 814. Subsequently a bookmark is set to have the value of the first value read at 816. The bookmark can be employed to keep track of the values that have been examined and in general where the process is in its examination of the partition. At 818, the count is incremented. The count is incremented so as to reflect the fact that a value has been read in and copied to the bookmark. Furthermore, since no other values have been read in yet the first value is unique or distinct. Next, another value is read from the partition at 820. Assuming that the partition contains more than one value, this new value is then compared with the bookmark at 822. The comparison between the bookmark and the newly read value seeks to determine whether the value is another unique value or the same value as in the bookmark. Because the data in a partition is ordered according to an aspect of the subject invention prior to execution of the distinct count query, if there are multiple instances of a value they will appear in sequence. Thus, if the newly read value is equal to the value stored in the bookmark then the value is a duplicate and should not be counted. The process can then proceed to 824 where a determination is made as to whether the end of the partition has been reached. If there are no more values to analyze because the end of the partition has been reached then the count is returned at 832. Thus far only one value has been uniquely read so the return count would be

one. If the end has not been reached, then the process proceeds to 820 where another value is read from the partition. If, however, at 822 the bookmark did not equal the newly read value then the count can be incremented at 826 to indicate the presence of another unique value. Subsequently, the bookmark will be updated to hold the value of the newly read value at 828. Next, a check is made to determine whether the end of the partition has been reached at 830. If yes, then the count value is returned at 832. Returning the count value at 832 can refer to, *inter alia,* notifying a primary processor of the distinct count of the partition, for example, if the partition is an independent partition executed concurrently or simultaneously with other partitions. Alternatively, returning the count can refer to providing a client with the distinct count value is only a single partition is being queried. If the end of the partition is determined not to have been reached then the next value is retrieved at 820. The loop continues until the all values have been scanned by the method. As can be gleaned from the above description, methodology 800 is a very efficient manner of calculating distinct count. Such efficiency results in part from the fact that the data is preordered so that duplicative values follow in sequence. However, it should be appreciated that the disclosed method is exemplary and that those of skill in the art may recognized upon reading this specification various different methodologies or algorithms that are also to be considered within the scope of the present invention.

Please replace the paragraph on page 16 line 24- page 17 line 30 with the following amended paragraph:

Turning to Fig. 9, an exemplary distinct count query methodology 900 is illustrated in accordance with yet another aspect of the present invention. The exemplary methodology illustrates one manner in which partitions can be examined in parallel. It should be appreciated that for purposes of clarity and understanding the methodology is only concerned with two partitions, however such methodology could be extended to deal with many more than just two partitions. Furthermore, to further simplify the methodology it is assumed that at each partition has at least one value. At 910, the distinct count is initialized to zero to indicate that no distinct values have yet been detected. At 912, the first values from each of two partitions are retrieved. The partition with the lowest first value will be the first partition to execute, accordingly it will

be referred to as partition A while the other partition is partition B. At 914, bookmarks are instantiated and initialized with the values retrieved from respective partitions. Thus, BKMRK A corresponds to partition A and BKMRK B corresponds to partition B. Subsequently, at 916, partition A is examined to determine if there are any more values to be read that have not already be read an analyzed. If there are no more values to read from A then the process goes to 936 to determine if there are any more values to be examined from partition B. If, however, there are more values in partition A then the process proceeds to 918, 924, and 930 where BMRK A is compared to BMRK B. At 918, is BMRK A is not equal to BMRK B then the count is incremented once to indicate the existence of a single distinct value at 920. Then, at 922, BMRK A is set to the next different value in partition A, for instance by reading values and comparing them to the bookmark until the value is different. At 924, if BMRK A is equal to BMRK B then the count is incremented at 926 and at 928, the value BMRK A is replaced with the next different value, if available, of partition A and the value of BMRK B is replaced with the next different value in partition B, if available. If, at 930, BMRK A is greater than BMRK B then the counter is incremented twice at 932 to account of the unique value in partition A and the unique value in partition B. Subsequently, the value of BMRK B is replaced with the value of the next different value in partition B at 934, if available. Then if the bookmark values fall all the way though 930 the process proceed to 916 to determine if all the values in partition A have been accounted for and evaluated. If no, then the process loops again through 918, 924, and 930. If yes, the process goes to 936 where it is determined whether all the values in partition B have been accounted for and evaluated. If yes, then the count is returned at 942. If no, then the count is incremented, at 938, and BMRK B is assigned to the next different value in partition B, at 940. The loop will then continue until all values in partition B have been evaluated. Subsequently, the count will be returned at 942. It should be noted and appreciated returning the value at 942 might not be the final distinct count as a count from one or more independent partitions running concurrently therewith might have to be added to the count.